## A problem

means an objective or an output that we have to achieve, we have to follow consecutive steps sequentially to attain the required objective

## Problem solving

The problem is defined as the objective or the specific output that we want to attain; through a sequence of steps and activities and, specific input.

## Problem solving stages

### 1 - Problem Definition

implies the identification of required outputs, available inputs and, arithmetic and logical operations to be executed.

### 2 - Performing step-by-step instructions (Algorithm) to solve a Problem

a group of logically arranged procedures to be executed to attain a goal or precise output, out of specific inputs.

a plan in the form of a series of successive steps is made
which is called an (Algorithm)
devised by the Mathematician and the founder of Algebra “Muḥammed ibn Mūsā al-Jwārizmī”.
The algorithm is represented by drawing “Flowcharts”

### 3 – program design

Having drawn a “Flowchart”, to solve the problem, using a computer; we have to translate this flowchart into one of the programming languages.

## 4 – program testing

we begin entering data to the program with previously known results; to compare the results of the current program to those of the well-known results; therefore we check the errors and debug them.

## 5 – program documentation

- Writing all steps taken for solving the problem that include: given Input, output
- plan for solving the problem, drawn flowchart
- programming language used for coding and, instructions
- date of last modification of the program and, people who - contribute to the program development process
- to have the program documented to go back
- for feedback and correction the documentation is beneficial when more than one person participate in writing or modifying the program

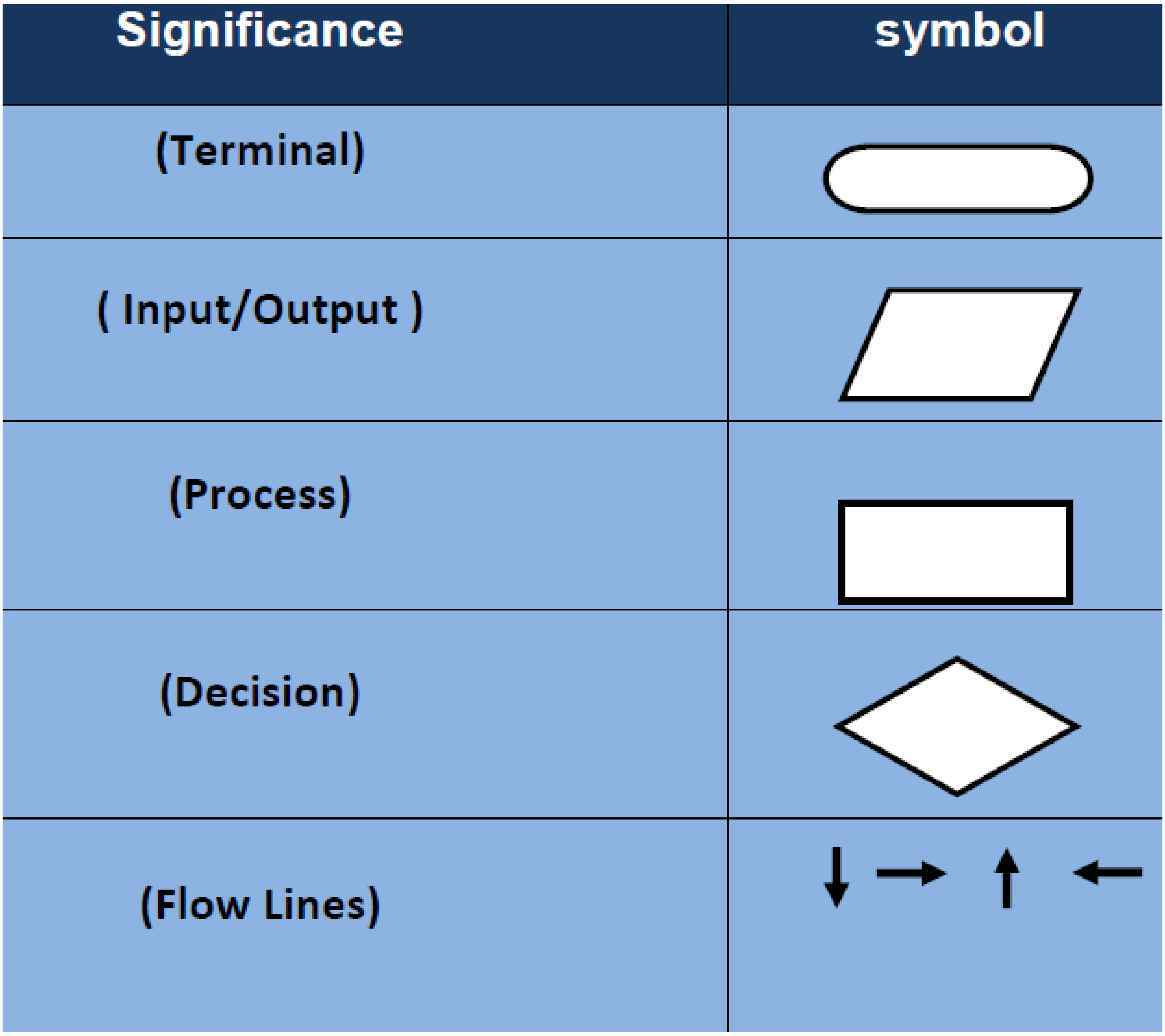
## Flowchart

is a diagram that uses standard graphical symbols; to illustrate the sequence of steps required for solving a problem or specific question.

- Flowchart promotes understanding of a problem
- shows what tasks should be performed when writing program codes
- so coding becomes an easy task for a programmer.
- A Flowchart explains the program to others

- it is also considered a convenient tool for documenting a program especially if it is complicated.

**The most commonly used symbols are as follows**



| Significance | symbol |
|---|---|
| (Terminal) | ⬭ |
| ( Input/Output ) | ▱ |
| (Process) | ▭ |
| (Decision) | ◇ |
| (Flow Lines) | ↓ → ↑ ← |

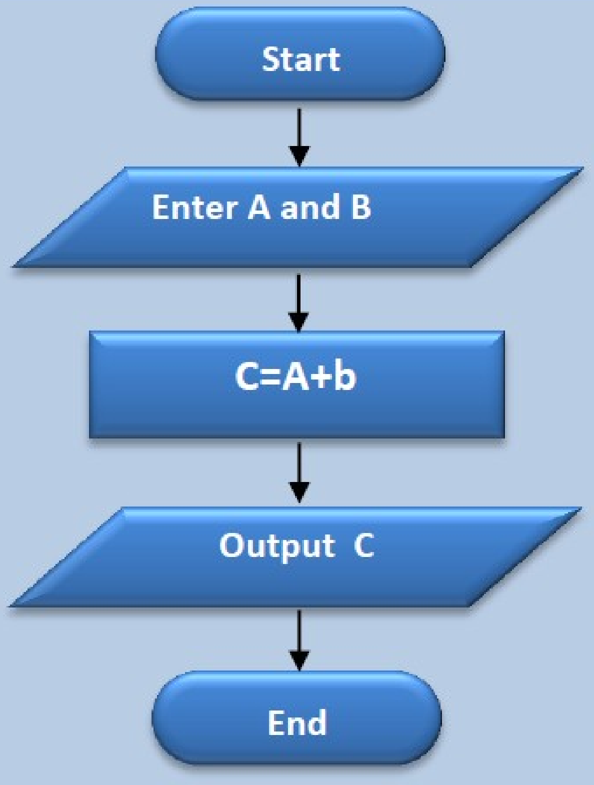
# Simple Flowchart

## Exercise 1

Draw a flowchart for a program that will calculate the sum of two numbers entered by the user and display the result.

**First**: Define the problem

**Output**: The sum of two numbers

| Second :Algorithm | Third :Flowchart |
|---|---|
| 1 Start<br>2 Enter the number A and the number B<br>3 Performing the sum of the two numbers using this equation C=A+B, the output is C<br>4 Print C<br>5 End |  |

## To construct a Flowchart we should consider the following

1-The flowchart should start with the Start symbol and end with the End ⬭ symbol.

2-A,B,C are variable names .A Variable refers to a memory storage that holds a value.

3-The equation: C =A+B, indicates the sum of the value of A, to the value of B, and stores the result in C.

4-Entering values in A and B is done by using the term "Enter", inside the parallelogram, ▱ you can also use another term to get the same meaning like "Read" or "Input".

5-The sum equation is written inside the rectangle, ▭ as it represents an arithmetic operation.

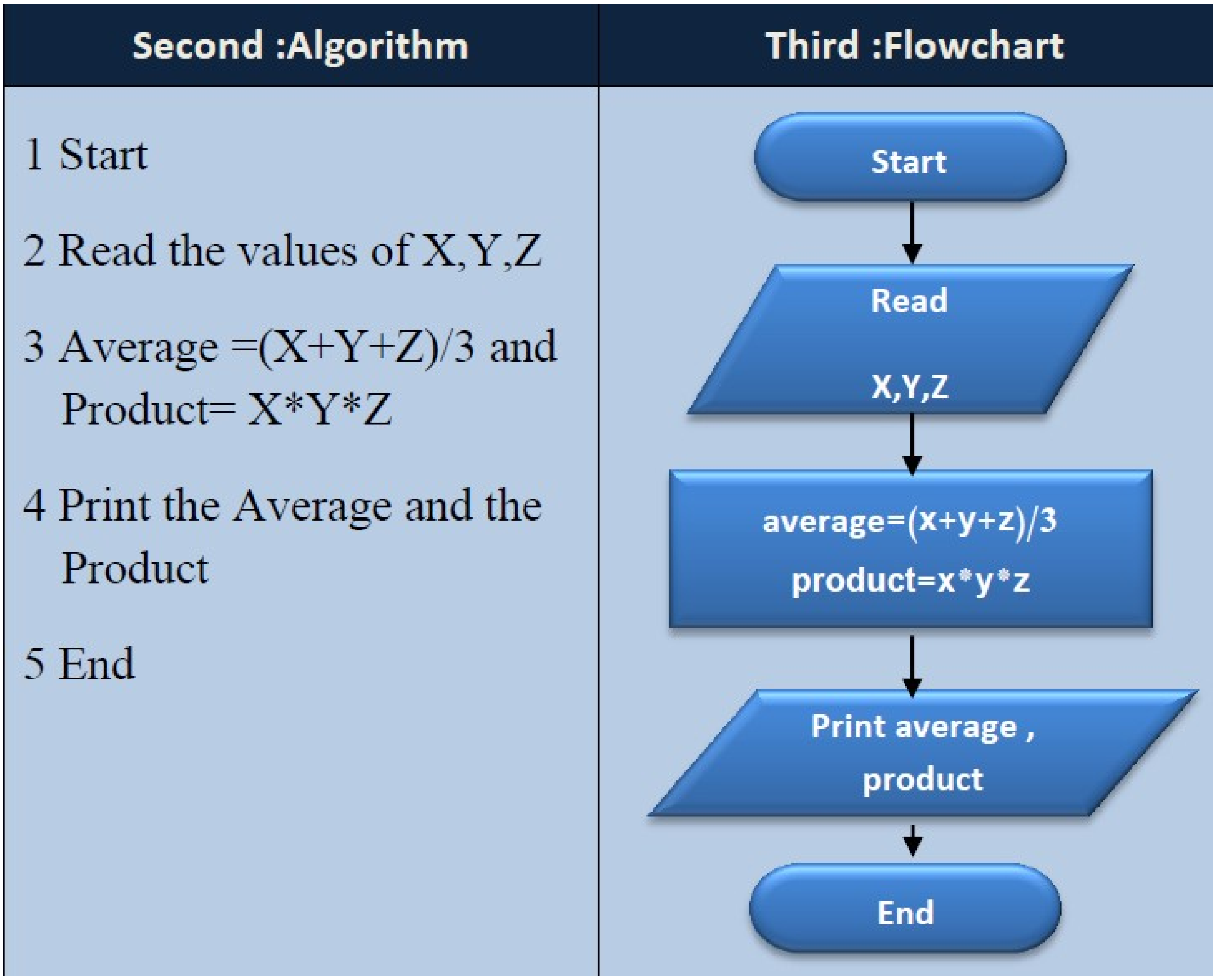
6-The output is expressed with a parallelogram ▱ using the term "Output", we can also use another term like "Print".

7-Note that Lines with arrows (flow lines) are from top to bottom ↓ and show the exact order of an Algorithm.

## Exercise 2

**Draw a flowchart for a program that will compute the average and Product of three numbers.**

**First**: Define the problem
**Output**: The average of three numbers.
**Input**: The number X, the number Y, and the number Z.
**Solution**: Average =(X+Y+Z)/3 and, Product= X*Y*Z.



| Second :Algorithm | Third :Flowchart |
|---|---|
| 1 Start<br>2 Read the values of X,Y,Z<br>3 Average =(X+Y+Z)/3 and Product= X*Y*Z<br>4 Print the Average and the Product<br>5 End | Start<br>Read X,Y,Z<br>average=(x+y+z)/3 product=x*y*z<br>Print average , product<br>End |

## Exercise 3

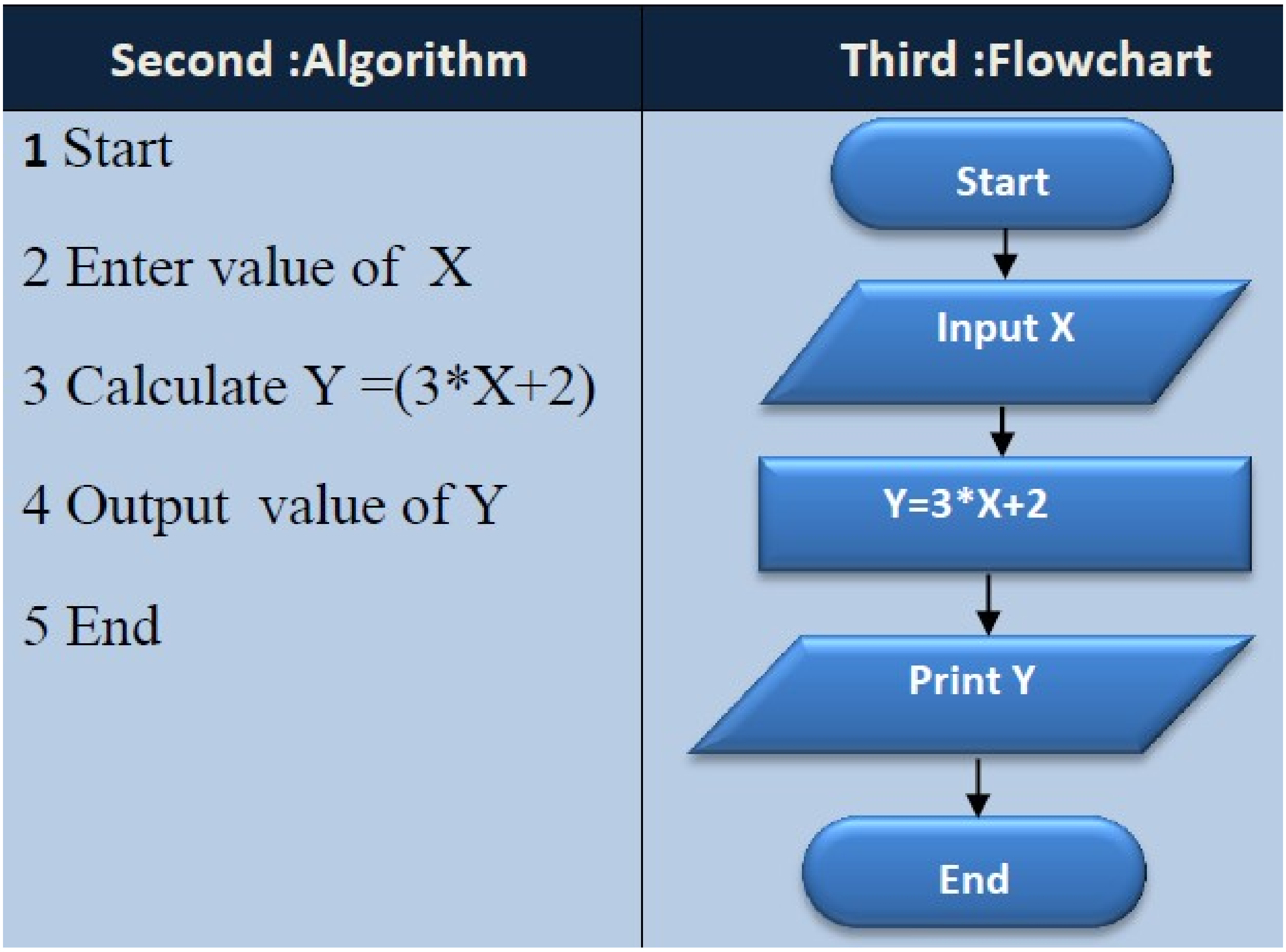
# Solving a first degree equation Y=3x+2

**First**: Define the problem

**Output**: The value of “Y”.

**Input**: X.

**Solution**: Compute the value of “Y” from the equation Y=3x+2



| Second :Algorithm | Third :Flowchart |
|---|---|
| 1 Start<br>2 Enter value of X<br>3 Calculate Y =(3*X+2)<br>4 Output value of Y<br>5 End | Start<br>Input X<br>Y=3*X+2<br>Print Y<br>End |

## Notice

1- The expression on the left side of any equation should contain only one variable; the value of this variable will be the (output) or the solution of the equation.

2- The expression on the right side of the equation may contain values or arithmetic expressions that have one or more variables (inputs).

3- Variable names are chosen to reflect and indicate the content of the variables

## Activity 1

Write down the Algorithm, and draw a flowchart to compute the area and the perimeter of a rectangle ,whose length and width are known , bearing in mind that the equation of the area is : Area =L*W and that of the Perimeter is: Perimeter =2*(L+W).

## Activity 2

Write down the Algorithm, and draw a flowchart to calculate the area of a circle whose radius “R” is known, bearing in mind that the equation of the area is: Area =3.14*R*R.

## Activity 3

Write down the Algorithm, and draw a flowchart to calculate the number of years, bearing in mind that the number of months is known

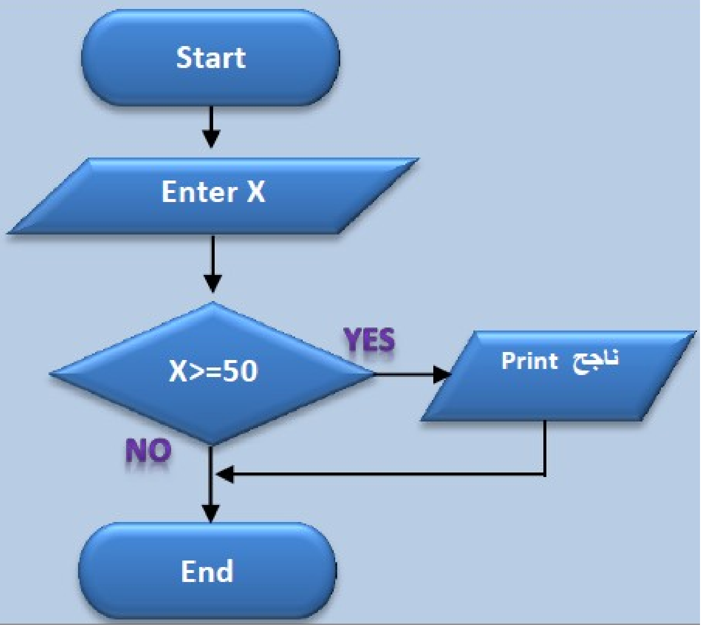
## The use of Branching (Decision) in Flowcharts

Most problems include a decision point, such as yes/no question, where two possible answers are available a “yes” and a “no”, in a flowchart the decision point has two paths (branches), each presents one of the possible answers, you can also find more than two answers.

### Exercise 4

Draw a flowchart for a program that will obtain exam scores from the user. Determine whether the score is greater than or equal 50 and display the message “ ناجح“.

**First**: Define the problem
**Output**: print the word “ ناجح “.
**Input**: the score X.
**Solution**: If the value of X is greater than or equal 50; the word “ناجح ” will be printed.

| Second :Algorithm | Third :Flowchart |
|---|---|
| 1 Start<br>2 Enter the values of X<br>3 If X> =50 then<br>3-1 Print “ ناجح ”<br>4 End |  |

## Exercise 5

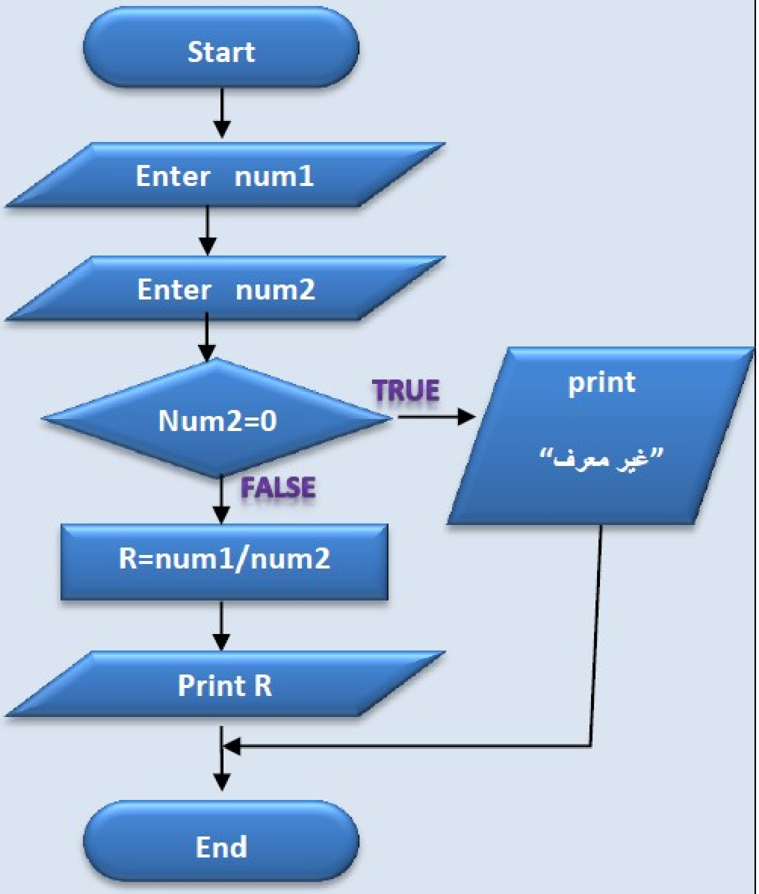
**Draw a flowchart for a program that will calculate the division of two numbers. Determine whether the divisor equal (zero) and display the message “unknown “.**

**First**: Define the problem

**Output**: print the result of dividing two numbers “R” or print the
word “غير معرف ”.

**Input**: the dividend is “num1”, and the divisor is “num2”.

**Solution**: if num2=0 then print “معرف غير”, otherwise print the
result of the division “R”.

| Second :Algorithm | Third :Flowchart |
|---|---|
| 1 Start<br>2 Enter the dividend num1<br>3 Enter the divisor num2<br>4 If num2 =0 then<br>4-1 Print “غير معرف”<br>4-2 Go to step 7<br>5 R=num1/num2<br>6 Print R<br>7 End |  |

## Exercise 6

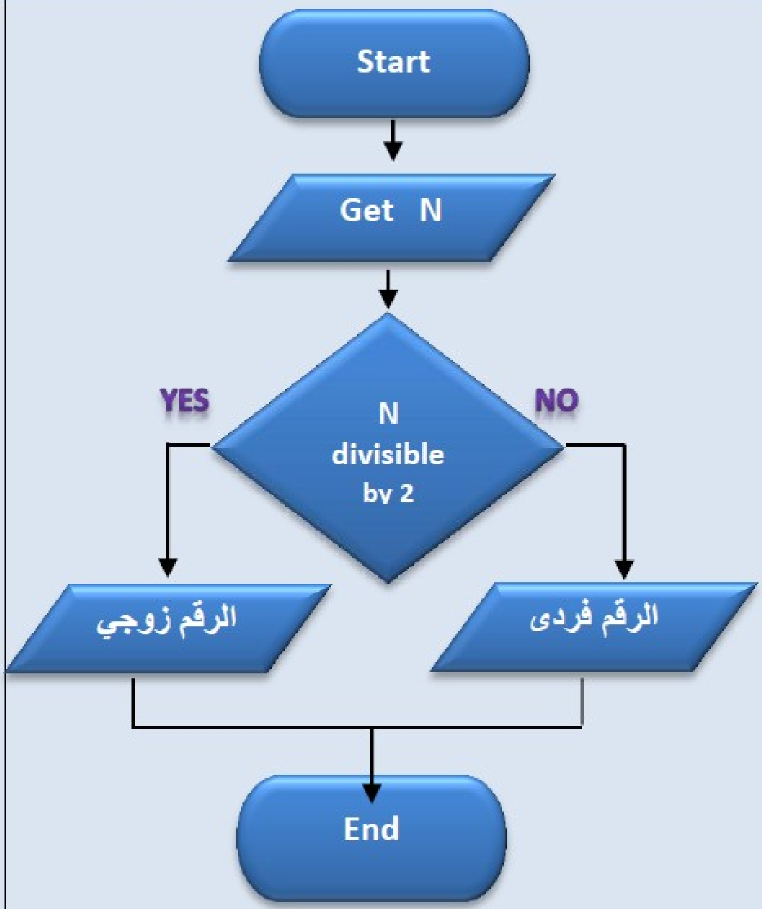
**Draw a flowchart for a program that obtains a number from the user.**

**Determine the number type (even or odd) and print the result**

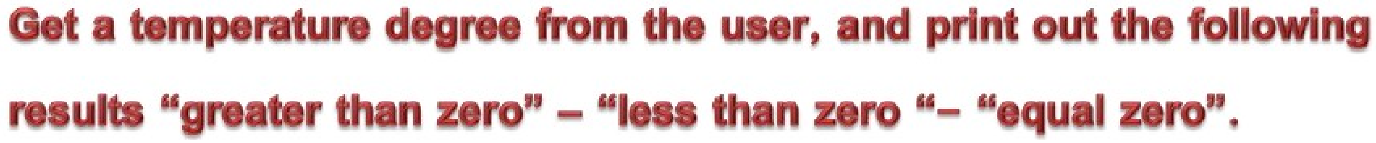
**First**: Define the problem

**Output**: print the number type (even or odd).

**Input**: the number “N”.

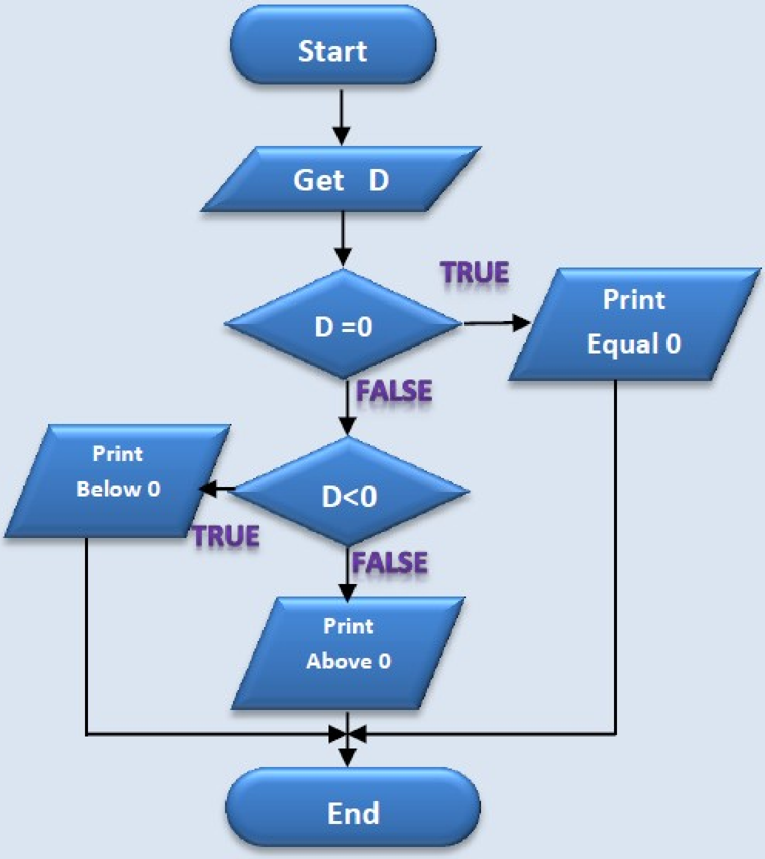
| Second :Algorithm | Third :Flowchart |
|---|---|
| 1 Start<br>2 Enter N<br>3 If N is divisible by 2 without remainder then<br>  3-1 Print “الرقم زوجي”<br>4 Else<br>  4-1 Print “الرقم فردى ”<br>5 End |  Start<br>Get N<br>YES<br>NO<br>N divisible by 2<br>الرقم زوجي<br>الرقم فردى<br>End |

## Exercise 7



Get a temperature degree from the user, and print out the following results “greater than zero” – “less than zero “– “equal zero”.

**First**: Define the problem
**Output**: print out “greater than zero” – “less than zero “-
“equal zero”.
**Input**: degree Celsius “D”.
**Solution**: the temperature degree entered will be compared to zero

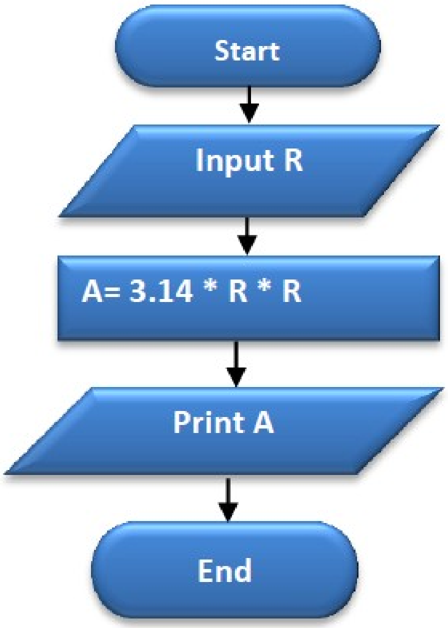
| Second :Algorithm | Third :Flowchart |
|---|---|
| 1 Start<br>2 Enter D (temperature degree)<br>3 If D=o then<br>3-1 Print “Equal zero”<br>4 Else<br>4-1 if D<0 then<br>4-1-1 Print “Below zero”<br>4-2 Else<br>4-2-1 Print “Above zero”<br>5 End |  Start<br>Get D<br>D =0<br>TRUE<br>Print Equal 0<br>FALSE<br>D<0<br>Print Below 0<br>TRUE<br>FALSE<br>Print Above 0<br>End |

## Activity 4

Write down the Algorithm, and draw a flowchart to enter two numbers, then Print “the largest is …? “ and, “the smallest number is…? “.

## Activity 5

The following flowchart is used to calculate the Area of a circle whose radius “R”. Repeat drawing the Flowchart so that it displays the message “not allowed “and exits from the program (When the value of “R” is negative).

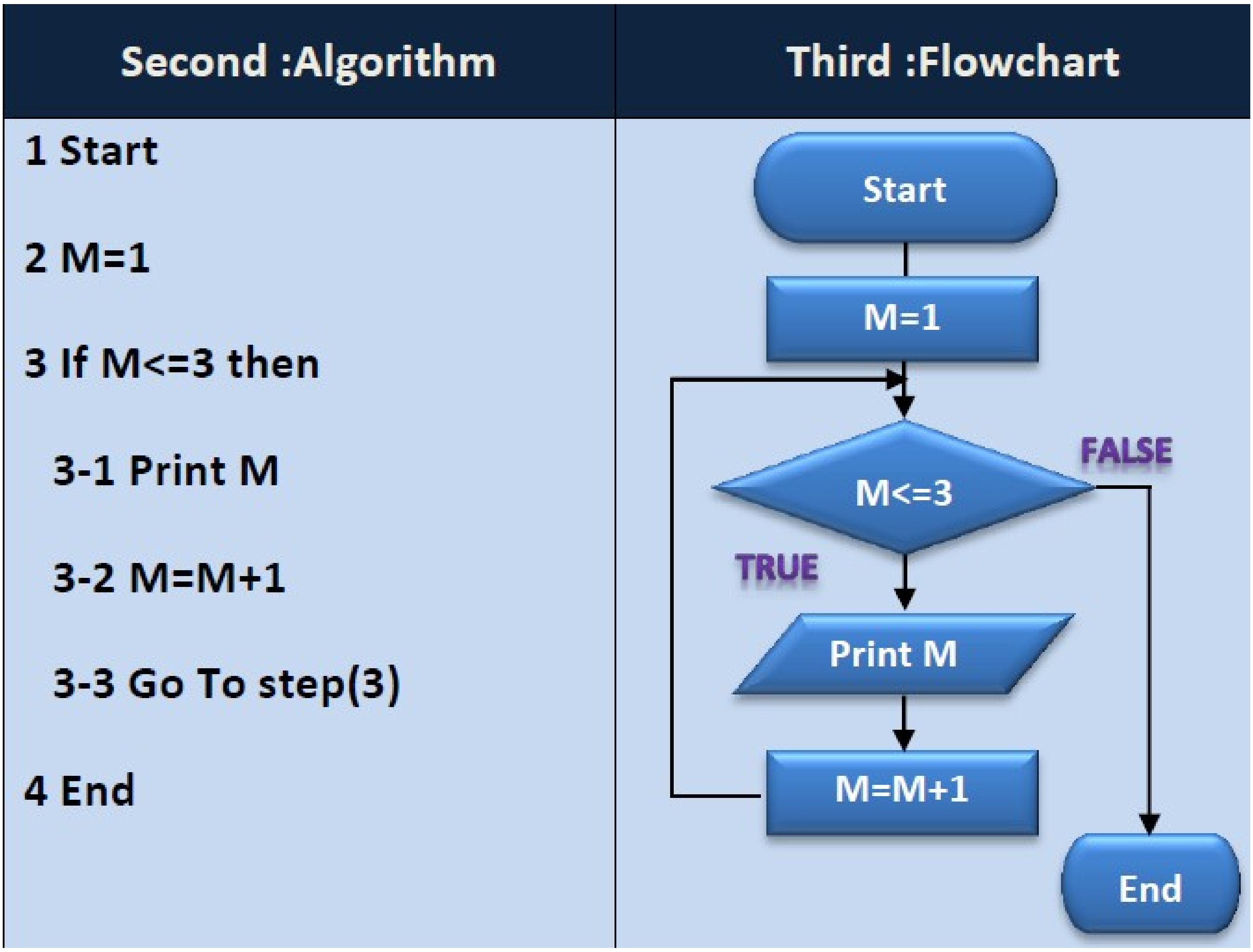
| Flowchart | Modified Flowchart |
|---|---|
|  | |

# The use of Loop in Flowcharts

## Exercise 8

**Print out the numbers from 1 to 3.**

**First**: Define the problem
**Output**: print out the numbers from 1 to 3
**Input**: the number M
**Solution**: print out the number M and increase by 1 then print out until reach the value to 3



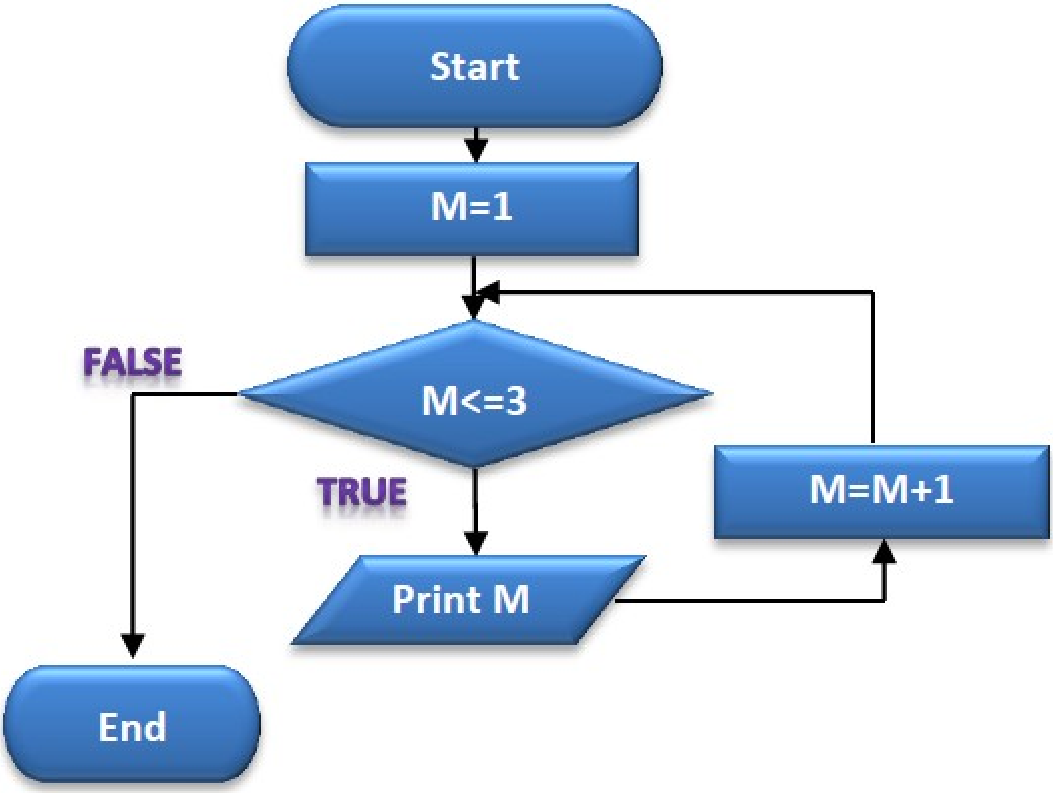
Second :Algorithm

```
1 Start
2 M=1
3 If M<=3 then
  3-1 Print M
  3-2 M=M+1
  3-3 Go To step(3)
4 End
```

Third :Flowchart

## A different way for drawing this flowchart

so it does not matter the figure drawn, what really matters i

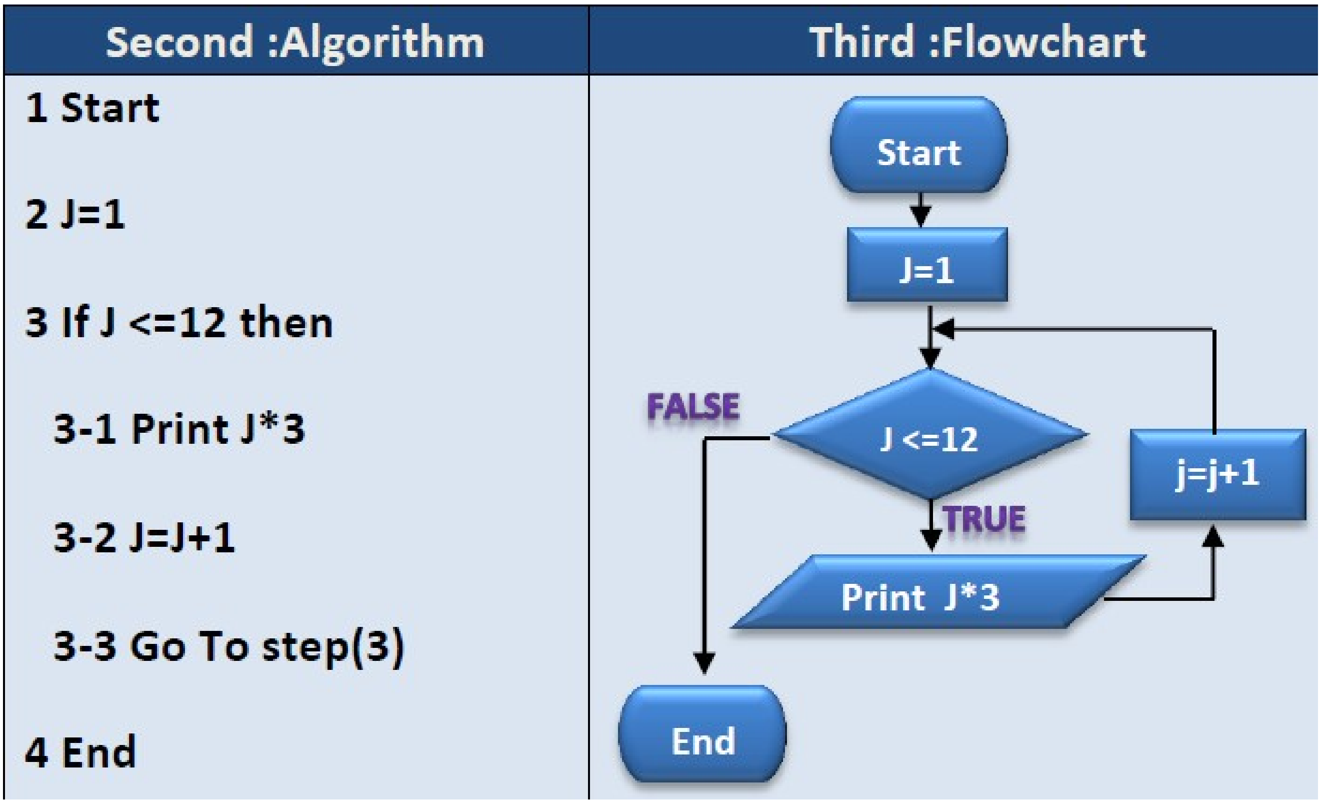


## Notice

1-M values that are printed 1 – 2 – 3
2- When the condition become false, the value of M that 4
3-M value that 4 after the end of the loop
4-Number of iterations is known in advance 3 times
5-M variable called counter, where it causes repeat steps

## Exercise 9

**Modify the Flowchart of the previous exercise to print the multiplication table of No 3.**



| Second :Algorithm | Third :Flowchart |
|---|---|
| 1 Start<br>2 J=1<br>3 If J <=12 then<br>3-1 Print J*3<br>3-2 J=J+1<br>3-3 Go To step(3)<br>4 End | |

## Activity 6

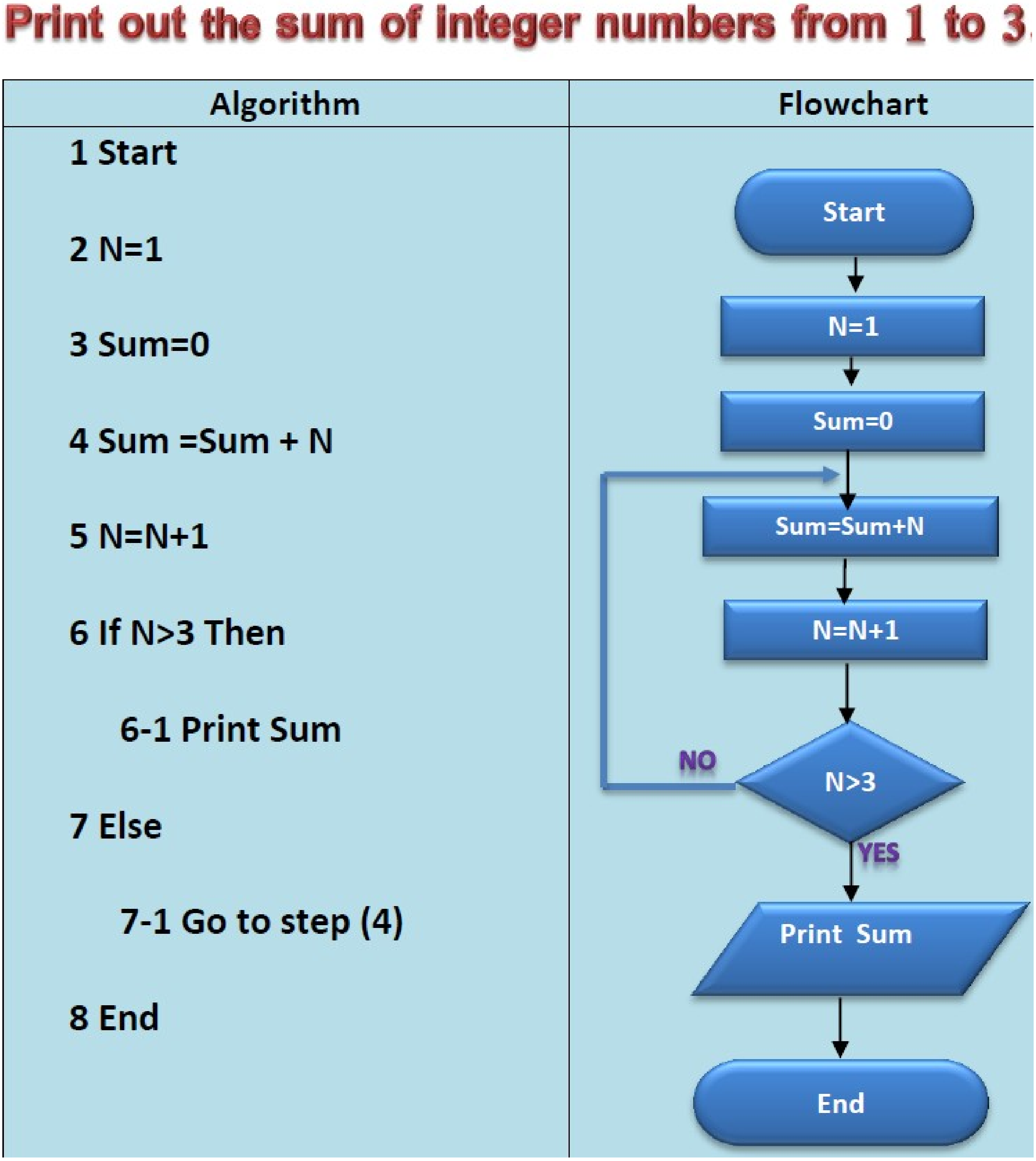
Track the values of the variable (J), and the printed value on executing every step in the previous exercise (write down in your notebook).
What is the value of the variable (J), when the result of the condition is “False”, and the loop ends?

**Activity 7**
Make the necessary modifications to the Flowchart of the previous exercise, so that you can enter the required multiplication table; instead of printing the multiplication table of “3” constantly

| Algorithm | Flowchart |
|---|---|
| 1 Start<br>2 Enter N<br>3 J=1<br>4 If J<=12 then<br>    4-2 Print J*N<br>    4-3 J=J+1<br>    4-4 Go To step(4)<br>5 End | |

| Second :Algorithm | Third :Flowchart |
|---|---|
| | |

## Exercise 10



Print out the sum of integer numbers from 1 to 3

Algorithm

```
1 Start
2 N=1
3 Sum=0
4 Sum =Sum + N
5 N=N+1
6 If N>3 Then
    6-1 Print Sum
7 Else
    7-1 Go to step (4)
8 End
```

Flowchart

## Notice

Variable N is considered a Counter, while variable Sum is an accumulating variable.

## Activity 9

Draw the Flowchart of the previous exercise in your notebook, after modifying it; to print out the sum of odd numbers from 1 to 10.

## Activity 10

After executing the previous activity, draw the flowchart in your notebook; to print out the sum of even numbers instead of the odd ones

# Chapter 2

## Introduction to Visual Basic.NET

with Visual Basic.Net Language, and you will be able to convert the solution steps of a problem to program codes that can be executed.

**Visual Basic .net**

Programming Language
is just one of the languages in Visual Studio .NET package
that includes other languages, such as C# and J#.
is an object-oriented language
that develops event driven Windows and Web applications.

**Programming Language**

Is a set of rules, symbols and special words you can use to write instructions and construct a computer program; according to the programming language used. Instructions will be translated to machine language for being executed.

**Notice**

A Computer executes only commands written in machine language. As for programmers; they can’t write in machine language, so they use programming languages to write programs in English; then comes the role of the compiler (found in the language) that translates program’ instructions from English to machine language; for a Computer to understand.

**Visual Basic .net**

is used to create windows applications
A Windows-based application has a Graphical User Interface (GUI) GUI appears in a window
like Paint , Notepad ,Calculator ,Internet browser that share some common characteristics
such as “Window style, Maximize button, Minimize button and, saving or opening files

**Windows applications are event driven applications**

you do an action (event) like pressing a plus (+) sign in the calculator application or from the keyboard, then a specified task will be executed, so writing programs using programming languages (as mentioned before) is important for responding to certain event.

**Visual Basic .net is an Object Oriented Language**

In Visual Basic.Net everything depend on Objects
(like: Button, Textbox, ComboBox)
which have
1- Properties that describe the Object

2- Events that occur to objects
3- Methods that present actions to be performed on objects ; causing certain behavior on the objects

**Example Calculator**

each button presents an (Object)
it has (Properties) like: width, height, text labeled, background color
has events like: (Click)
has method certain action does this button

**Object**

is the basic constructive element in Object Oriented Programming; it is created from a defined class.

**Class**

is the blueprint/ plan / template, from which the individual objects, are created.
Sets in (  Properties, Methods and, Events) which takes any object  from the (Class).

**Notice**

1.The (Class) implies a definition for the (Object).
2.The (Object) exists only when an instance of the class is created
3.You can create as many objects you need from a class.
4.A place in the memory is reserved for each object in Visual Studio.NET when it is created.

**.Net Framework**

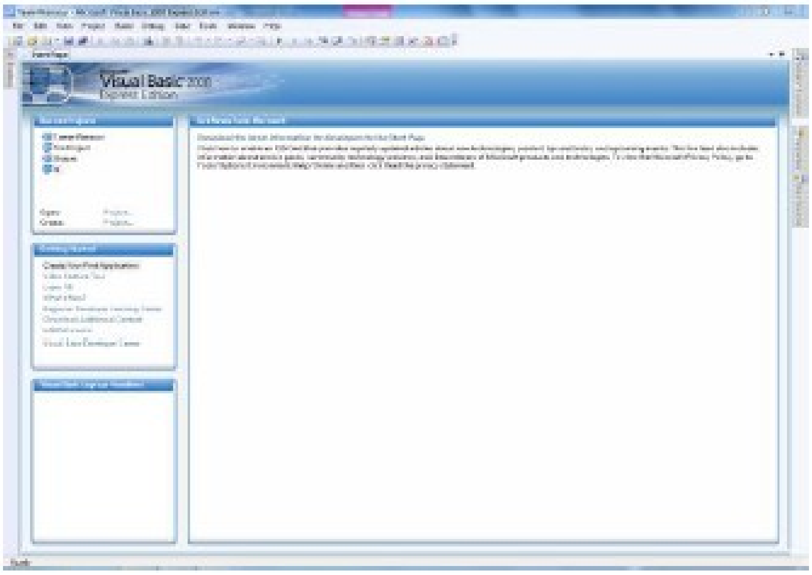
is like the central nervous system for all Visual Basic.Net applications, is used in the development, design and execution of .NET applications, can be installed for free on operating systems (and is available with many versions).

**that enables you to:**
1- Develop applications like (Desktop applications) - (Web applications) – (Mobile applications).
2- Provide a development environment for running all applications.

**The Framework is composed of:**

1. The execution engine (CLR) Common Language Runtime
2. The .NET class libraries (System Class Libraries)
3. (Compilers)
4. Other elements

**Main elements of IDE screen**
integrated development environment (IDE)
that enables the developer to do as much as possible with visual tools, to quickly design applications (Windows applications) - (Web applications) – (Mobile applications)



**Form**

The form is the window (visible interface) of the application
what users will see and work with when they run this application
a form is the container upon which controls (CommandButton –Textbox- Label) are placed

## Toolbox

contains controls (objects) that the programmer can place on the form, these controls are available in tabs (categories)
can display all the (Controls), by choosing (All Windows Forms) can display (Common Controls)
can display (Menus & Toolbars)

## Some common control tools

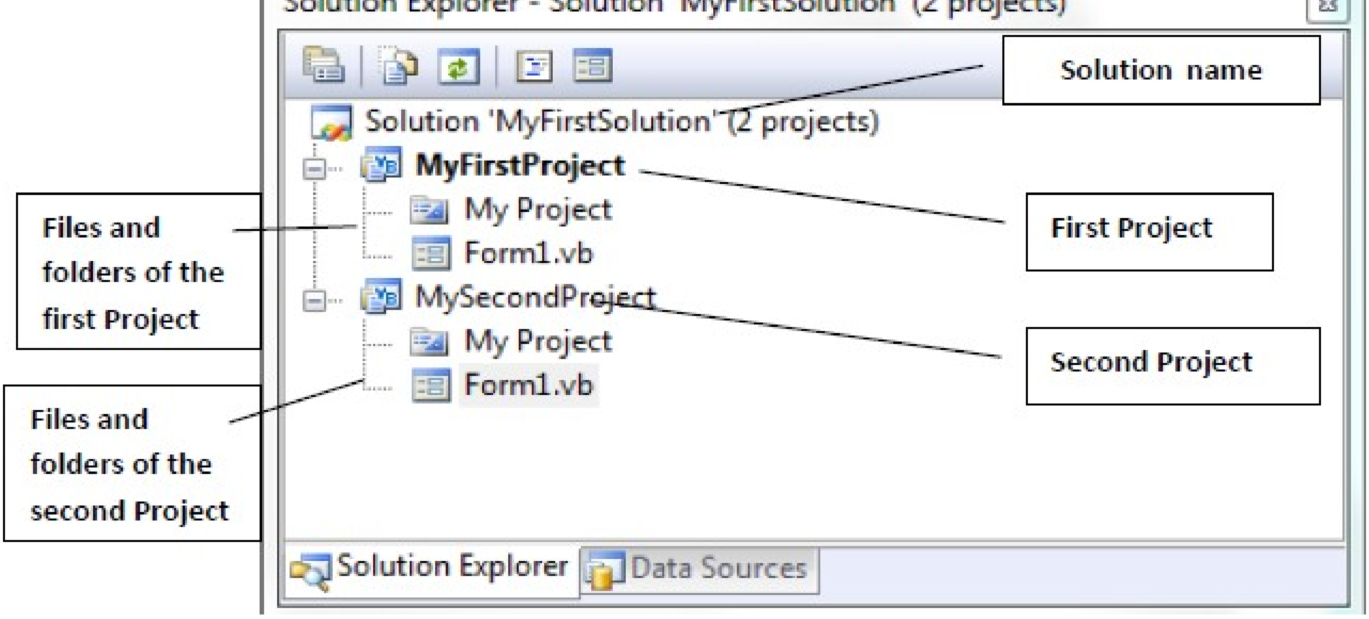
Button – TextBox – Label – ListBox – Combobox – checkbox – Radiobutton

## Properties Window

Each control from control tools has properties, can be set through the Properties window

## Solution Explorer

contains ( files and folders) for one or multiple projects of the current solution

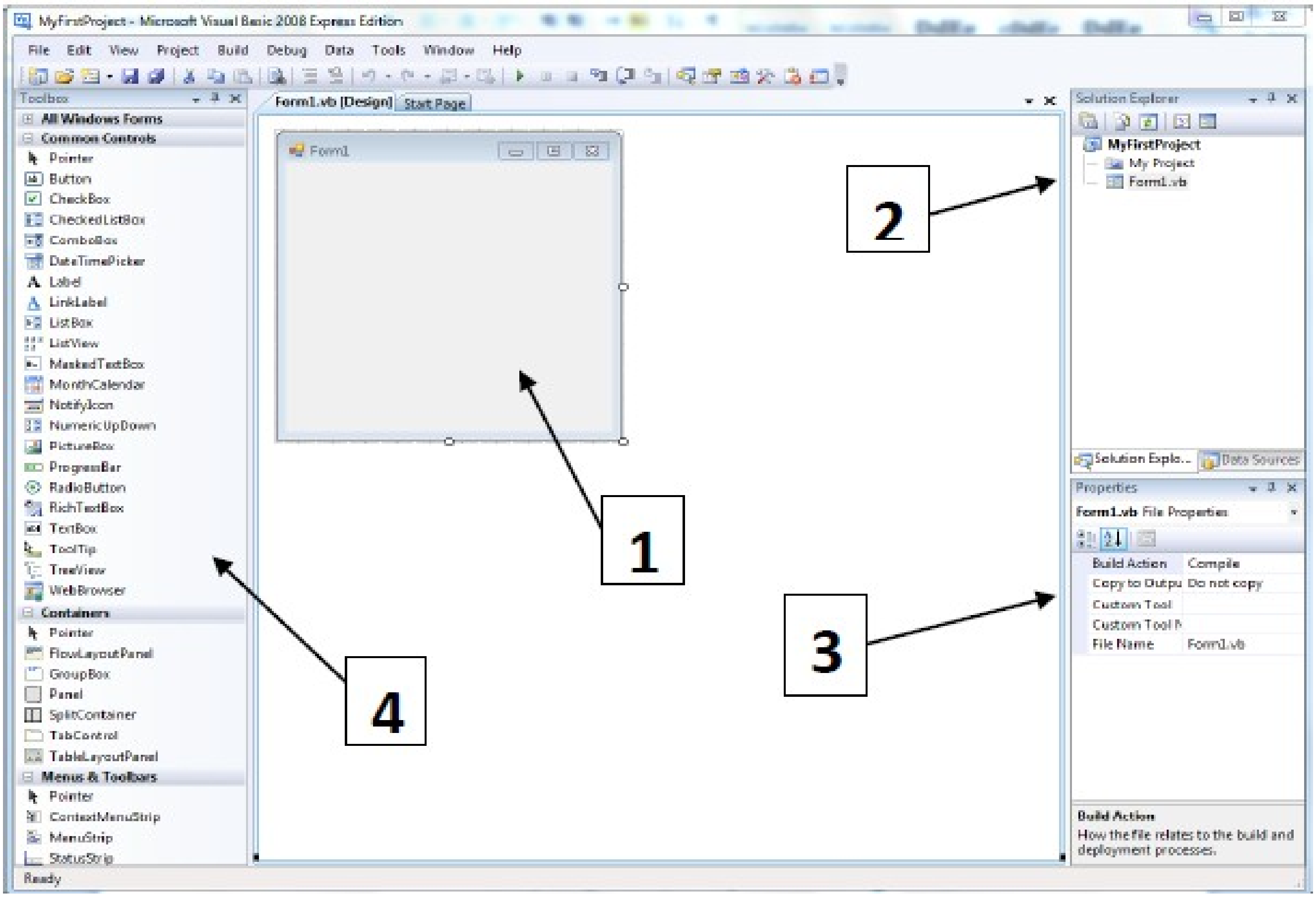

Solution Explorer - Solution 'MyFirstSolution' (2 projects)
Solution 'MyFirstSolution' (2 projects)
MyFirstProject
My Project
Form1.vb
MySecondProject
My Project
Form1.vb
Solution Explorer
Data Sources
Solution name
First Project
Second Project
Files and folders of the first Project
Files and folders of the second Project

## Eexrcise 1

Create a new project

1. From (File) menu choose (New Project) .
2. Choose template “windows forms application “
3. Type project name, then press ok

The IDE window appears



(1) The Form window
(2) The Solution Explorer Window
(3) The Properties window
(4) The Toolbox window

## Exercise 2

Add a new form to the project

1-From the (Project) menu choose (Add Windows form)

2-A window appears, choose template “ windows Form”
3-Type new form name, then press add

**Exercise 3**

Save the project in one of the storage devices
1-Choose (File) menu then select (Save All )
2-A window appears, choose storage devices, you can change project name which previously was written when creating a new project
3-press the (Save) button

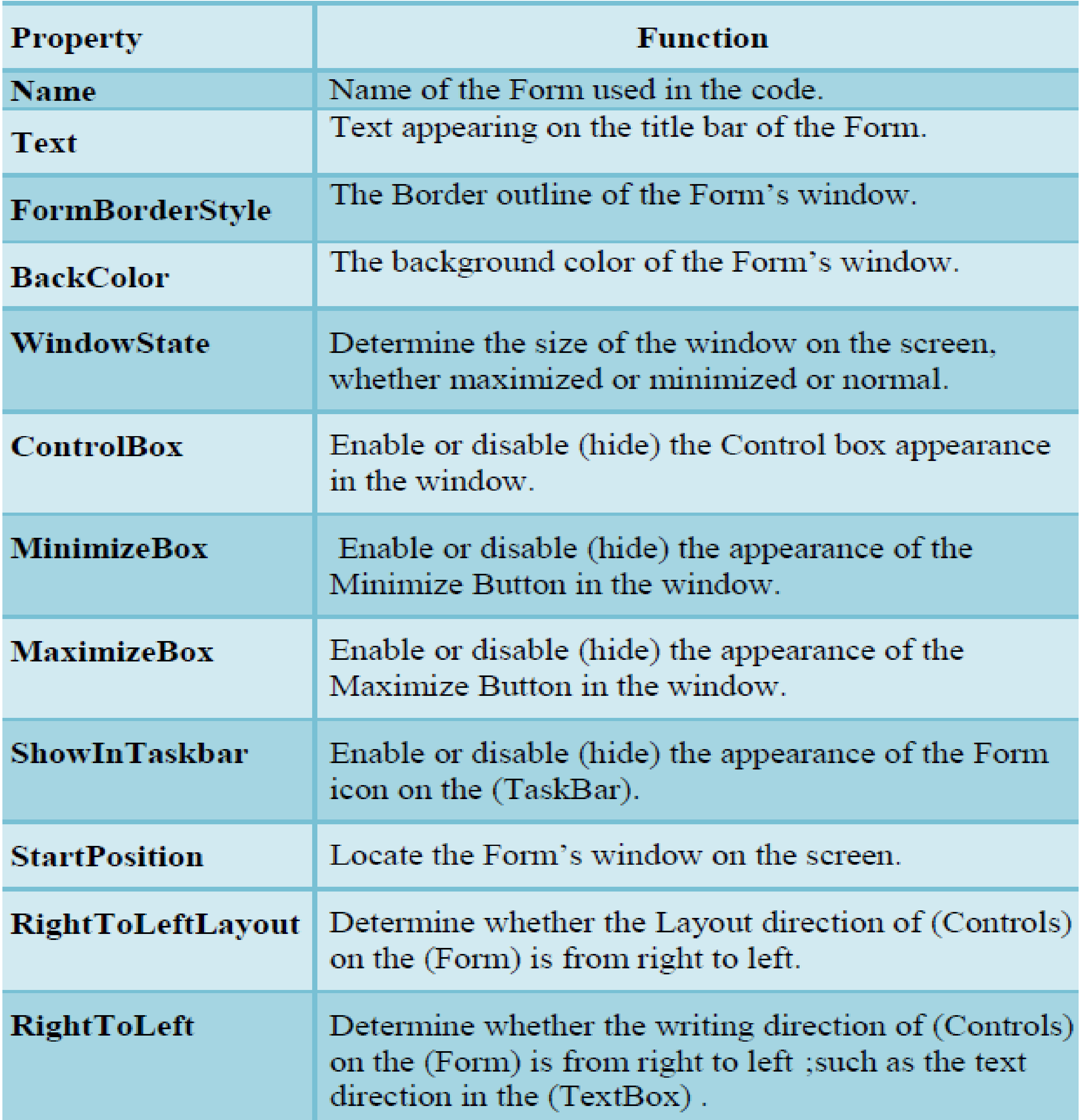
**Exercise 4**

Add a new project to the solution
1-Choose (File) menu, then select add, then select new project
2-A window appears, choose template, type new project name, then press ok

# Chapter 3
# Controls

## Notice

- Some properties (such as: Text – Name – Forecolor – BackColor – RightToLeft.) are common to most controls.
- Some properties will not be applied to controls placed on a form; unless we set other properties to



| Property | Function |
|---|---|
| **Name** | Name of the Form used in the code. |
| **Text** | Text appearing on the title bar of the Form. |
| **FormBorderStyle** | The Border outline of the Form's window. |
| **BackColor** | The background color of the Form's window. |
| **WindowState** | Determine the size of the window on the screen, whether maximized or minimized or normal. |
| **ControlBox** | Enable or disable (hide) the Control box appearance in the window. |
| **MinimizeBox** | Enable or disable (hide) the appearance of the Minimize Button in the window. |
| **MaximizeBox** | Enable or disable (hide) the appearance of the Maximize Button in the window. |
| **ShowInTaskbar** | Enable or disable (hide) the appearance of the Form icon on the (TaskBar). |
| **StartPosition** | Locate the Form's window on the screen. |
| **RightToLeftLayout** | Determine whether the Layout direction of (Controls) on the (Form) is from right to left. |
| **RightToLeft** | Determine whether the writing direction of (Controls) on the (Form) is from right to left ;such as the text direction in the (TextBox) . |

**Notice**

-the default values of the properties (Text) and (Name) are (Form1)

-the value of property (Text) show during design and running

- the value of (Text) property is displayed as text in the title bar of the Form

- does not show the value of the property (Name) during design and running

Because it used when writing the code in the code view

**Notice**

-The value of property (RightToLeft) is (Yes - no )

- The value of property (RightToLeftLayout) is (true - false )

- The property (RightToLeftLayout) will not be active, unless the property value of (RightToLeft) is (Yes).

**Notice**

-the Properties (controlbox – minimize – maximize) take true - false

-formborderstyle property has many values like sizable that Means the possibility of controlling the size of the form window through its borders
**-**if formborderstyle property take none value, Form borders disappear and the address bar also disappears

**Notice**
- ( ShowInTaskbar ) property take ( true – false )
- (StartPosition ) property has many options like ( centerscreen)
- (WindowsState) Property take ( maximize – minimize – normal )

-properties ( formborderstyle – controlbox – minimizebox – maximizebox – righttoleftlayout – righttoleft – backcolor – text )
When you adjust any property of them, shows the impact on the form window immediately ( during design )

-properties ( showintaskbar – startposition – windowstate )
When you adjust any property of them, doesn't show the impact on the form window immediately ( during design ), but show the impact During the test program ( running program )

-running the program
press F5 on the keyboard
click start icon on the standard toolbar
select start debugging from debug menu

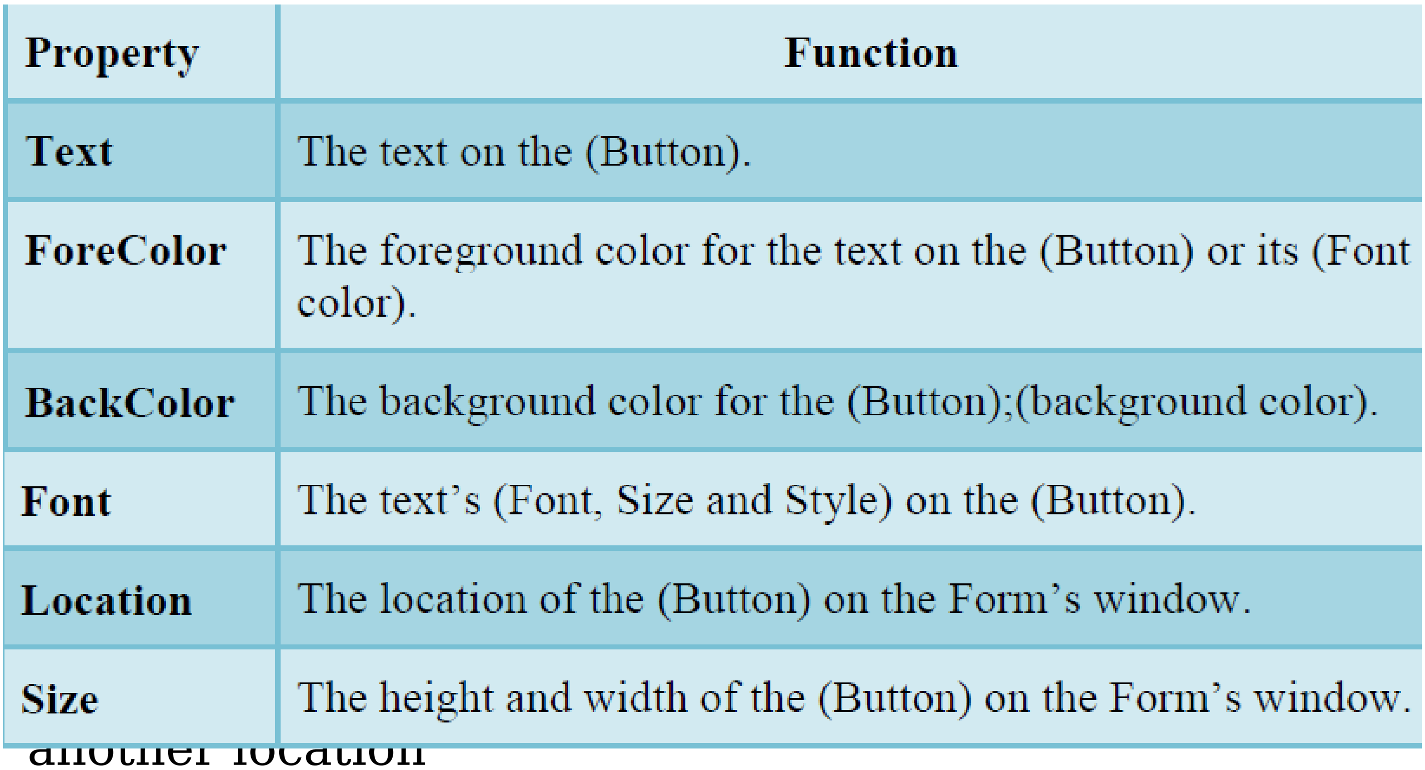
-stopping the program
Click button on the title bar of form
click stop icon on the standard toolbar
select stop debugging from debug menu

**Button**

A Button is one of the (Controls) that can be drawn on the (Form),
a user will use a button by clicking on it to perform a specific task

To place a Command Button on the form, in design mode. Move the mouse pointer to the Toolbox and double-click the Button icon
The Button is displayed on the Form

**Button Properties**



| Property | Function |
|---|---|
| **Text** | The text on the (Button). |
| **ForeColor** | The foreground color for the text on the (Button) or its (Font color). |
| **BackColor** | The background color for the (Button);(background color). |
| **Font** | The text's (Font, Size and Style) on the (Button). |
| **Location** | The location of the (Button) on the Form's window. |
| **Size** | The height and width of the (Button) on the Form's window. |

another location
Notice, change property value from 0;0 that default value to new value

**To adjust size property** active the button, Place the mouse pointer on one of the boxes (sizing handles). Drag and Drop the mouse until the desired size is reached

Notice, increase property value if pull out, but decrease if pull in

**Notice**

All the previous properties shows the impact on the button

***except name property***, The impact will not appear on the button and not on any other tool

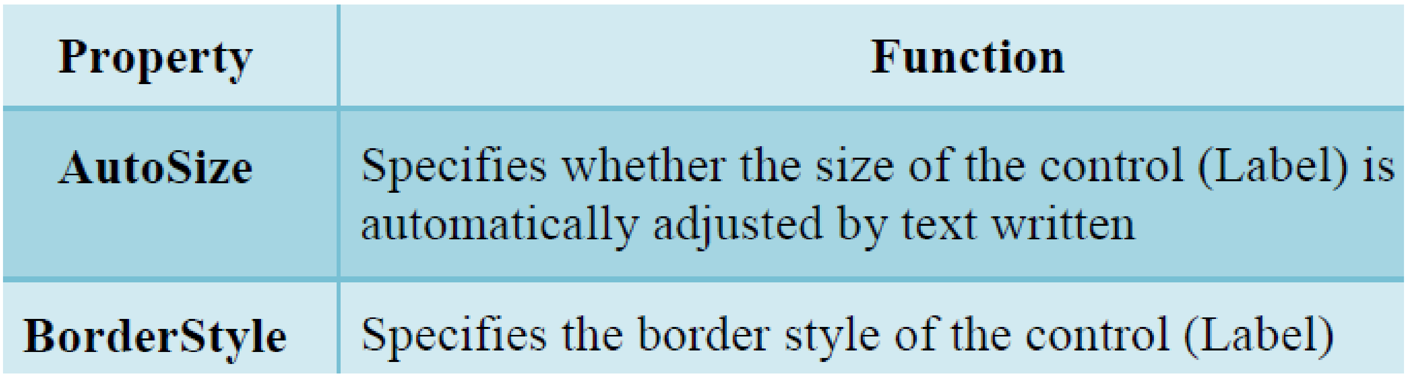
because this property, only used in the code view

## Label

is a control used to provide the user with information. It appears as a heading or title within a form to let the user know the form’s content.

Label controls cannot be changed, users cannot type in (any text) during the run-time.

## Label properties



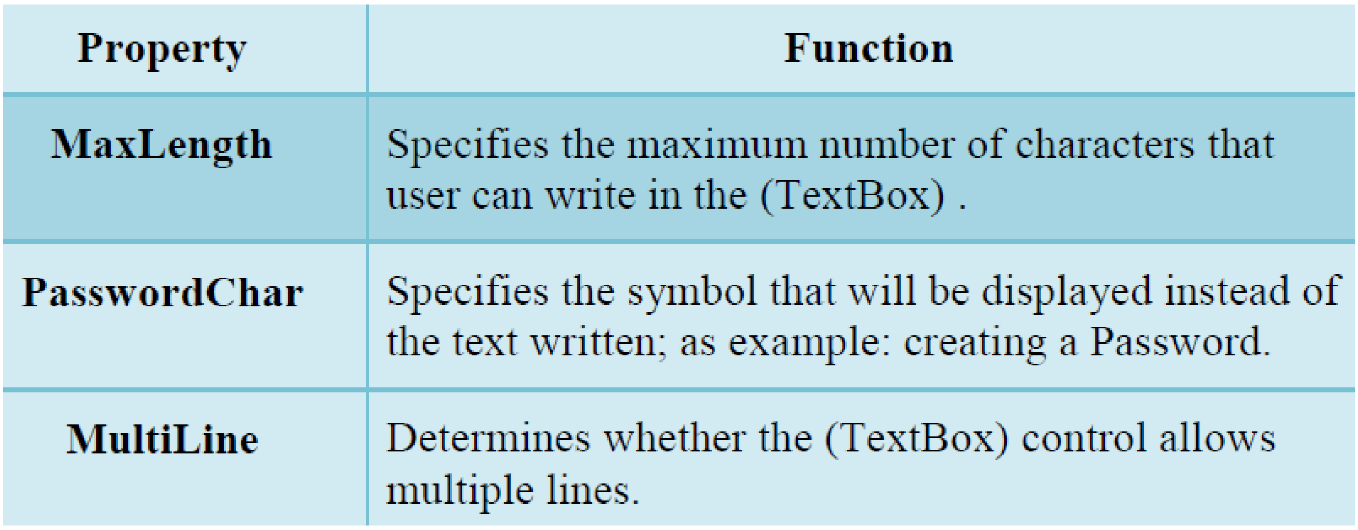
| Property | Function |
|---|---|
| AutoSize | Specifies whether the size of the control (Label) is automatically adjusted by text written |
| BorderStyle | Specifies the border style of the control (Label) |

**Notice**
If the AutoSize property is set to False, you can manually adjust the size of the label.
If the AutoSize property is set to True, the label size is automatically adjusted to fit the text displayed on the label.

**Notice**
The control (Label) has a set of properties like (Name- Text- Font –
ForeColor – BackColor – Visible – Size – Location – RightToLeft –
Image)

**TextBox**
can be used for both entering data and displaying results.

**TextBox properties**



| Property | Function |
|---|---|
| **MaxLength** | Specifies the maximum number of characters that user can write in the (TextBox) . |
| **PasswordChar** | Specifies the symbol that will be displayed instead of the text written; as example: creating a Password. |
| **MultiLine** | Determines whether the (TextBox) control allows multiple lines. |

**Notice**
Multiline property take ( true – false )
True means writing in Multiline
False means writing in one line

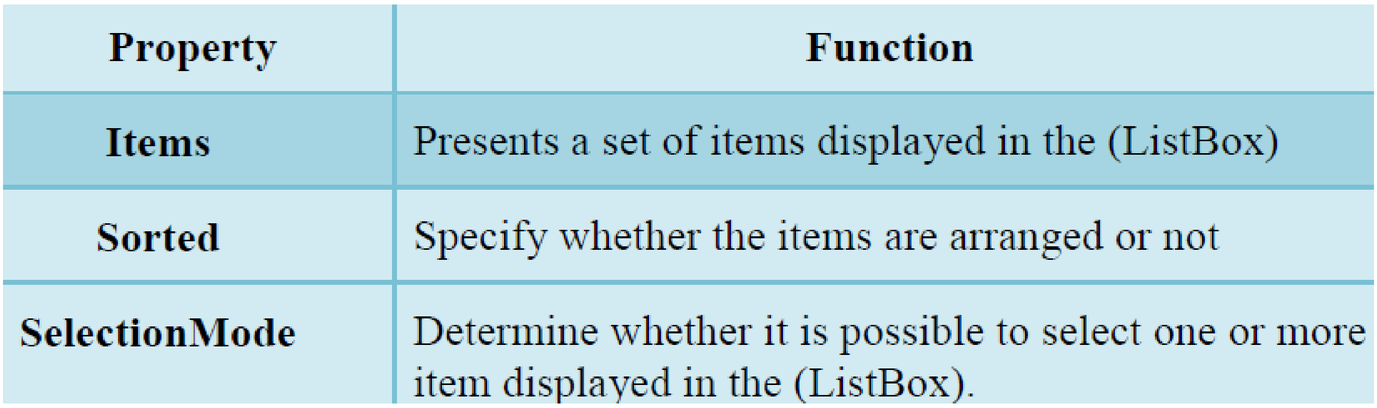
**Notice**

The control (TextBox) has a set of properties like (Name- Text- Font – ForeColor – Visible – Size – Location – RightToLeft – Enabled)

**ListBox control**

is used for displaying a list of items

## ListBox control properties



| Property | Function |
|---|---|
| **Items** | Presents a set of items displayed in the (ListBox) |
| **Sorted** | Specify whether the items are arranged or not |
| **SelectionMode** | Determine whether it is possible to select one or more item displayed in the (ListBox). |

**Notice**

Sorted property take true – false

selectionmode property take

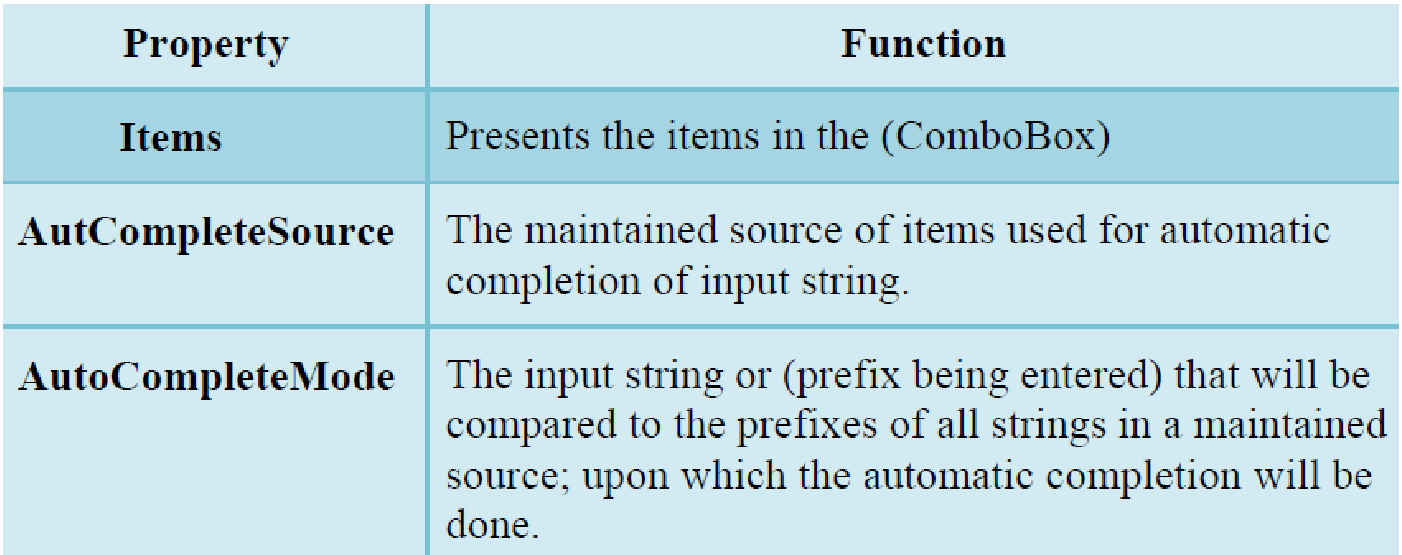
<u>one</u> To choose only one item

<u>multiextended</u> To choose more than one item

<u>none</u> Not be allowed to choose any item

**Notice**

The control (ListBox) has a set of properties like ( Name –Visible –Size– RightToLeft – ForeColor- Font -BorderStyle – location )



| Property | Function |
|---|---|
| **Items** | Presents the items in the (ComboBox) |
| **AutCompleteSource** | The maintained source of items used for automatic completion of input string. |
| **AutoCompleteMode** | The input string or (prefix being entered) that will be compared to the prefixes of all strings in a maintained source; upon which the automatic completion will be done. |

**Notice**
Autocompletesource property have several choices, including
Listitems items in the list
None Not choose from any source

**Notice**
Autocompletemode property have several choices, including
suggest Suggested items of the current list (By the beginning of writing user )
None Do not display any suggestions

**Notice**
The control (ComboBox) has a set of properties like ( Name –Visible – Size –Location – RightToLeft – ForeColor- Font -BorderStyle –Items )

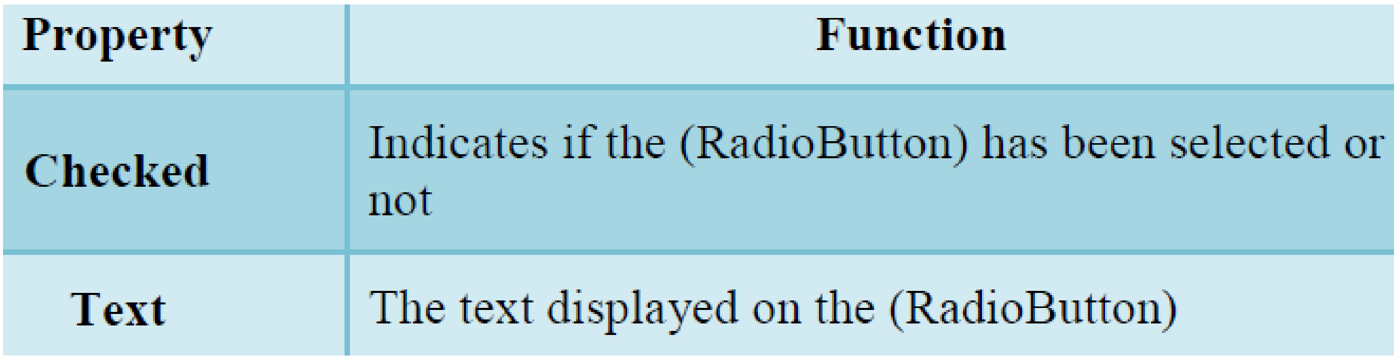
**GroupBox control**
is used to group other controls of same function together on the Form window.

**GroupBox control properties**
The control (GroupBox) has a set of properties like ( Name –Visible – Size –Location – RightToLeft – ForeColor- Font – text – background )

### RadioButton Control

is used to select one option only from a group of options

### RadioButton Control properties



| Property | Function |
|---|---|
| Checked | Indicates if the (RadioButton) has been selected or not |
| Text | The text displayed on the (RadioButton) |

**Notice**

The control (RadioButton) has a set of properties like ( Name –Visible – Size –Location – RightToLeft – ForeColor- Font)

**Notice**

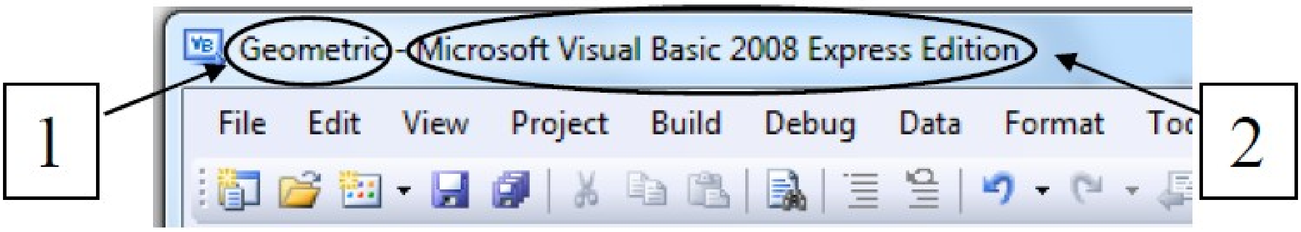
Checked property take true – false
To choose more than one alternative of radiobutton
Can work two group of groupbox  tool,  put a number of radiobutton in each group
Then,  can select  one radiobutton from groupbox, and select another radiobutton from another groupbox

### Checkbox control

is used to select one or more options

**Checkbox control properties**
The control (CheckBox) has a set of properties like ( Name –Visible – Size –Text – Checked – Font – ForeColor– RightToLeft -Location)

# Chapter 4
# Code Window



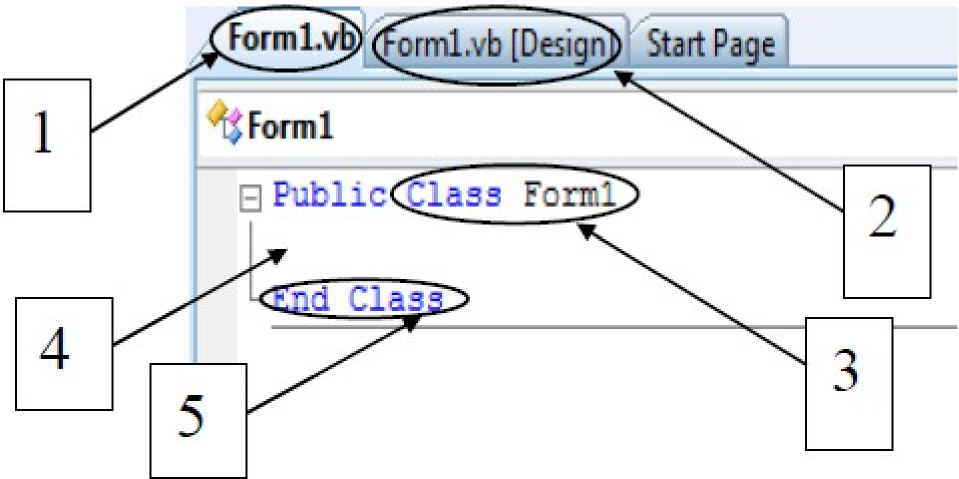
(1) The Solution name.
(2) (Visual Studio) version used.

## Code Window

Through the Code window; we can write instructions and codes using (Visual Basic .Net) language

**To open the (Code Window) of (Form1) perform the following:**

1- Make sure that the window Form is active

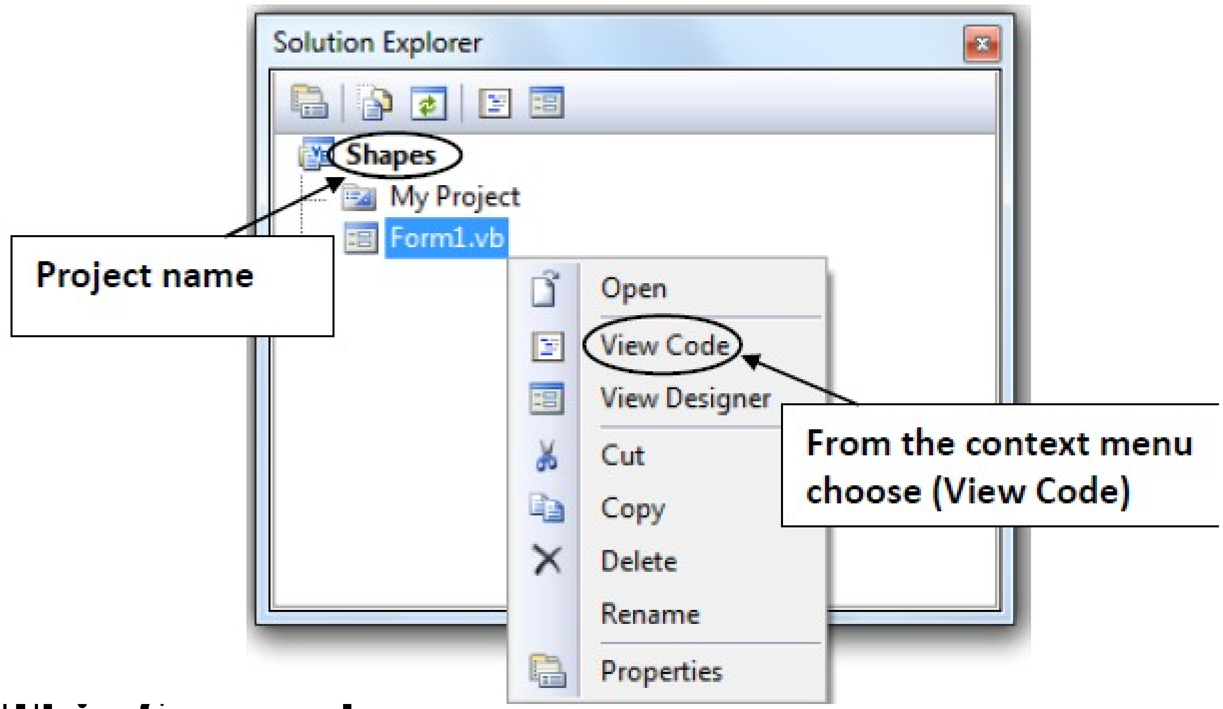


```
Public Class Form1

End Class
```

(1) Name of the file where codes are saved.
(2) Name of the file where the Form window is saved.
(3) The declaration of Class; its name is (Form1).
(4) Space between two lines; to type codes for the Class (Form1).
(5) The end of the Class.

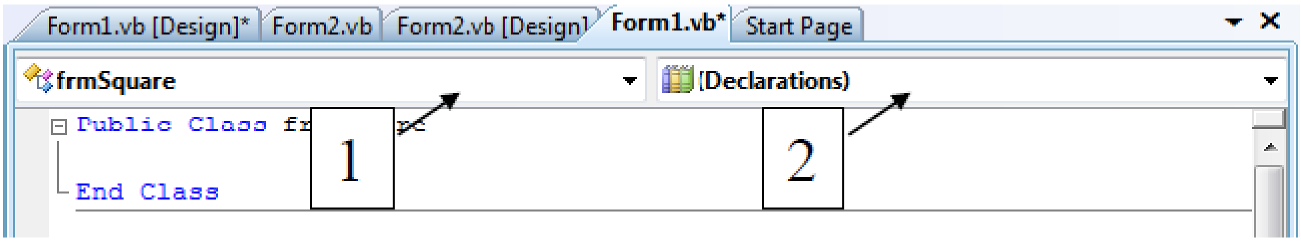
## Another way to open the (Code Window) of (Form1)



Project name

From the context menu choose (View Code)

## This figure shows

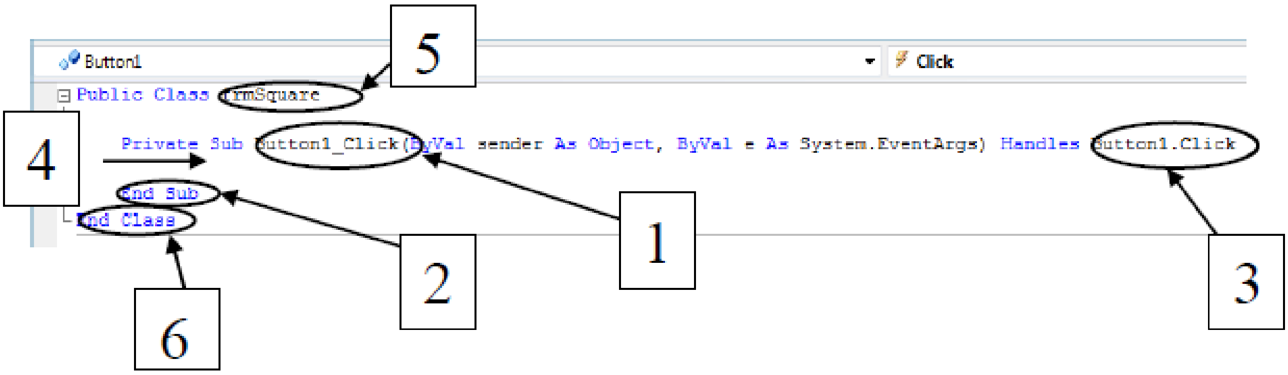


```
Public Class fr          re

End Class
```

1- A drop-down menu of (Class Names), which refers to the names of controls placed on the Form.
2- A drop-down menu of (Method Names) or events; associated with the Class selected from the (Class Names) menu
3- Open the drop-down (Class) menu and note that the default names of the controls are listed, that you put on this form
4- When you select (Button1) from the Class menu, open the drop-down (Methods) menu; it displays the events associated with (Button1)

**Event Handler**
is the procedure (called into action) when an event occurs



```
Public Class frmSquare
    Private Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
    End Sub
End Class
```

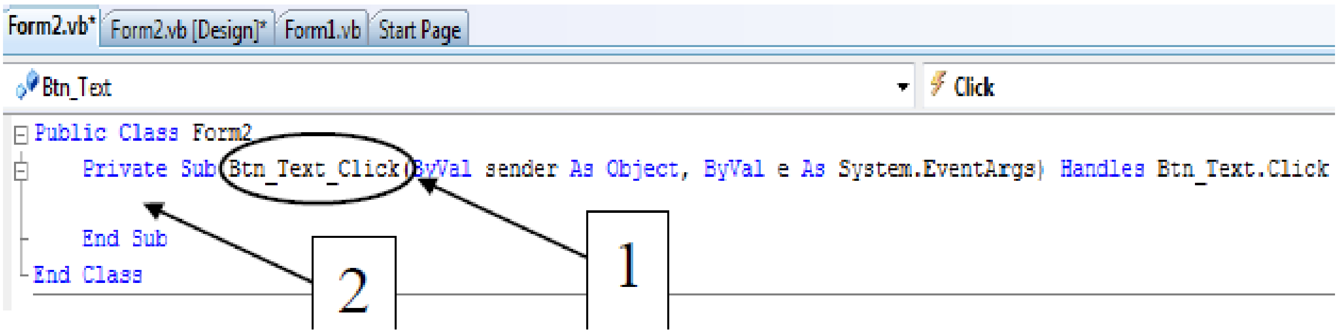
(1) The procedure name composed of (object name, event name)
(2) End of procedure

(3) What causes the call of the procedure (event occurrence)

(4) Between the two lines, you can write codes that will be executed, when call of the procedure, after event occurs

(5) The declaration of the Class (frmSquare)

(6) The end of the Class



```
Public Class Form2
    Private Sub Btn_Text_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Btn_Text.Click

    End Sub
End Class
```

1-The Procedure Name is (btn_Text_Click);
 the control name is (btn_Text), and the event name is (Click).

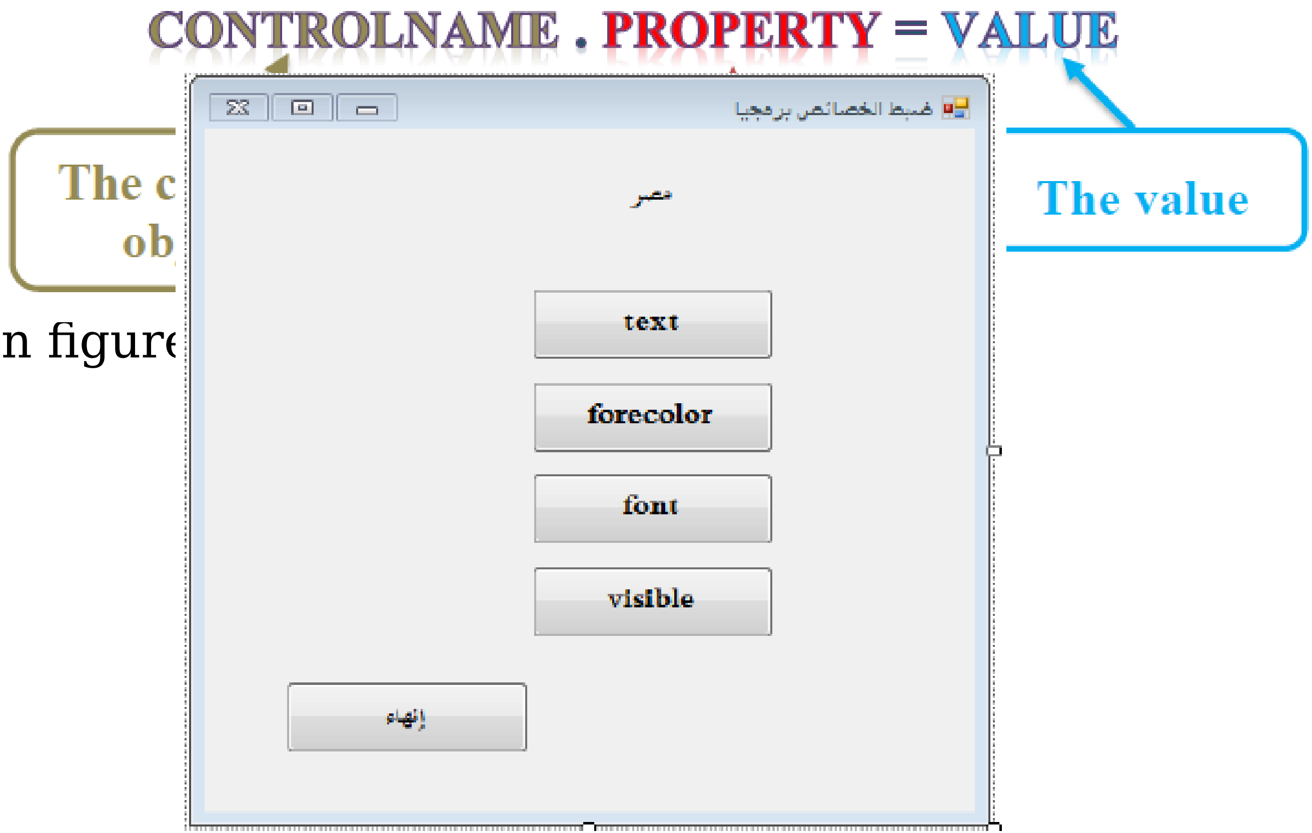
2-Between the two lines shown; you can write statements or codes.

## Setting the (Properties) programmatically

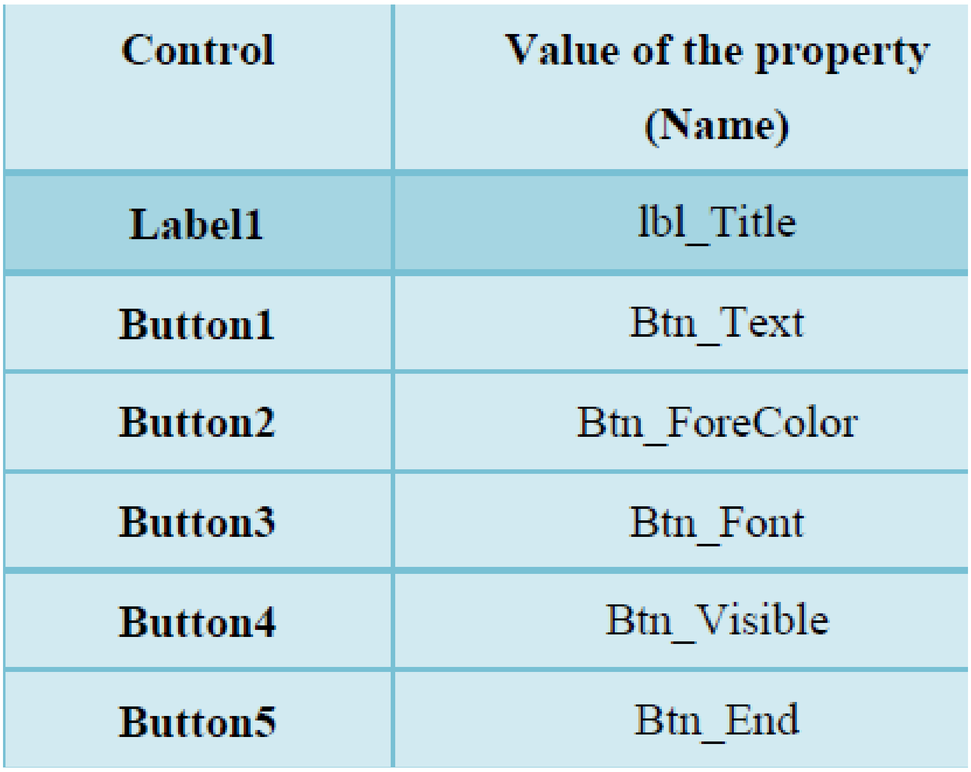
]

(

in figure



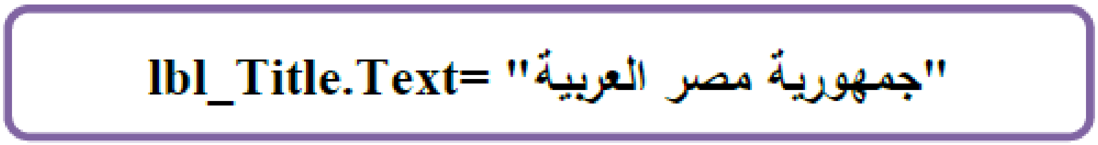
Set the value of the property (Name) for the controls as follows in table



| Control | Value of the property (Name) |
|---|---|
| **Label1** | lbl_Title |
| **Button1** | Btn_Text |
| **Button2** | Btn_ForeColor |
| **Button3** | Btn_Font |
| **Button4** | Btn_Visible |
| **Button5** | Btn_End |

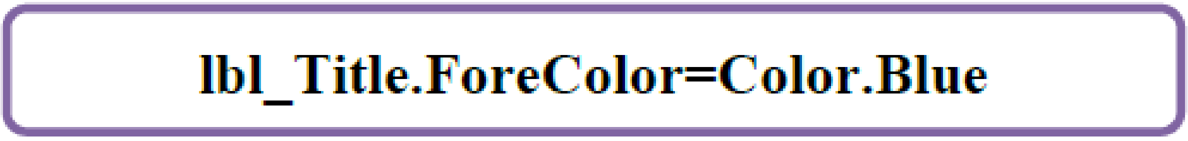
- Insert the (Click )event handler of the Button (btn_Text )
- Adjust the property (Text ) for the Label (lbl_Title ) to be :

  “ جمهورية مصر العربية “



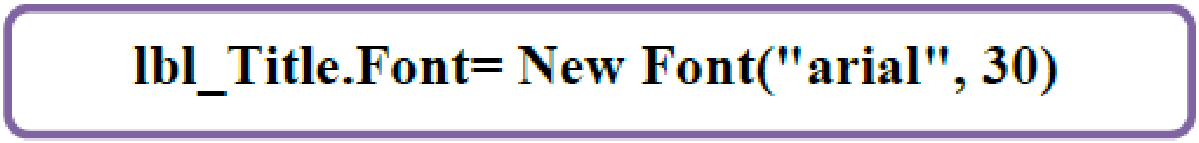
```
lbl_Title.Text= "جمهورية مصر العربية"
```

-Insert the (Click )event handler for the Button (btn_ForeColor)
- Set the value of the property (ForeColor) for (lbl_Title ) to (Blue)



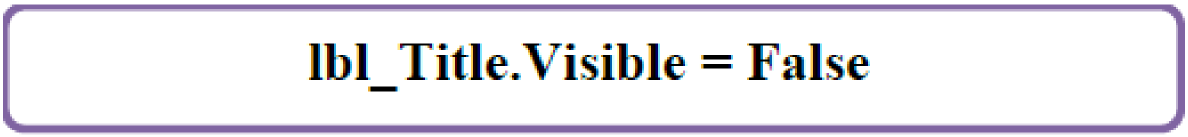
```
lbl_Title.ForeColor=Color.Blue
```

-Insert the (Click )event handler for the Button (btn_Font )
- Adjust the property (Font) for (lbl_Title ) to become: (font type) = Arial and, (font size) =30



```
lbl_Title.Font= New Font("arial", 30)
```

-Insert the (Click )event handler for the Button (btn_Visible)
- Set the value of property (Visible) for (lbl_Title ) to (False)



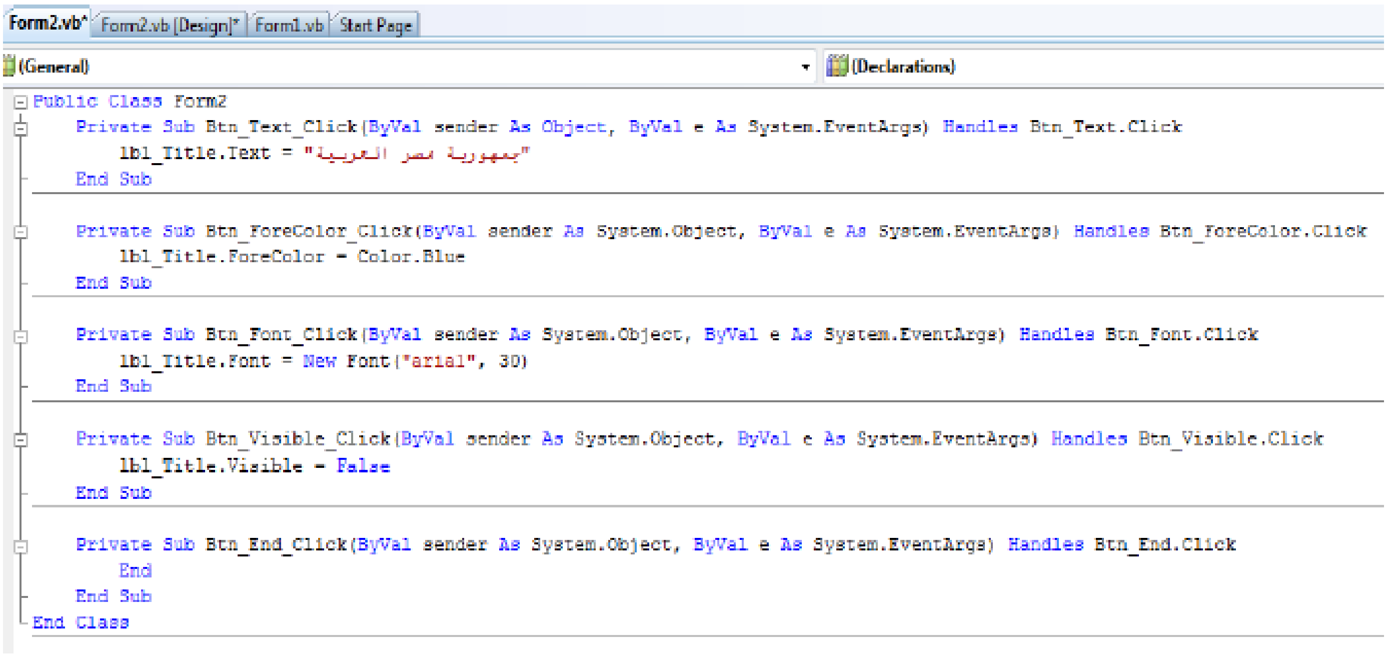
```
lbl_Title.Visible = False
```

**Notice** The values assigned to properties may have several types:
1.The abstract value; e.g. Property (Text).
2.The logical value; e.g. Property (Visible).
3.The value selected from a list; e.g. Property (ForeColor).

4.The value formed from creating an (Object); e.g. Property (Font).
5.The value of the result of arithmetic expression (will be studied later).
6.The value of a Variable or Property.

-To end the procedure insert the (Click) event handler for the Button (btnEnd) and type the command (End)

when we terminate writing all (Event Handlers). The (Code Window) becomes as show in figure



```
Public Class Form2
    Private Sub Btn_Text_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Btn_Text.Click
        lbl_Title.Text = "جمهورية مصر العربية"
    End Sub

    Private Sub Btn_ForeColor_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_ForeColor.Click
        lbl_Title.ForeColor = Color.Blue
    End Sub

    Private Sub Btn_Font_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_Font.Click
        lbl_Title.Font = New Font("arial", 30)
    End Sub

    Private Sub Btn_Visible_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_Visible.Click
        lbl_Title.Visible = False
    End Sub

    Private Sub Btn_End_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_End.Click
        End
    End Sub
End Class
```

With best wishes